REMARKS

The Examiner has rejected claims 1-5 and 8-9 under 35 U.S.C. 103 as unpatentable over Thangadurai (U.S. Patent 6,748,526) in view of Fish (U.S. Patent 6,381,693). The Examiner has rejected claims 6-7 under 35 U.S.C. 103 as unpatentable over Thangadurai (U.S. Patent 6,748,526).

The Amended Claims

Claim 2 has been amended to clarify that information from the processor type register must be read into a management processor of the system. Claim 2 has also been amended to clarify that the selected compatible boot image is mapped into memory space of the processor, and to provide indentifiability of the processor in dependent claims.

Claim 3 has been amended because claim 3 now inherits the management processor from its parent claim 2.

The New Claims

New claims 10-12 are dependent upon amended claim 2, and provide additional limitations.

New claim 13 essentially corresponds to the scope of former claim 8, written in independent form, with the additional restriction that the system be a heterogeneous computer system.

Processor Familes

In the microprocessor and microcontroller art, there are frequent references to processor families. Typically, an individual processor family includes processors having many basic instructions in common. While basic instructions are identical, members of a family often have differences in extended instructions, cache, memory and input/output devices provided on the chip.

Consider for example the Intel x86 family, including the 8086, 80286, 80386, 80486, Pentium, Pentium-2, and successor processors produced by Intel over the past 25 years. All members of this family have a basic set of common instructions for processing 8-bit bytes and 16-bit integers. Typically, at least one processor mode on

late-model processors allows for direct execution of the same binary encoded instructions that ran on the ancient 8086.

Members of the Intel x86 family differ, however, in that in addition to the basic instructions of the 8086, the 80386 has features for accessing a greater amount of memory than could be addressed by an 8086, the 80486 has extended instructions for use with an on-chip floating point processing unit and the memory controller of the 80386, and the Pentium machines add internal cache memory to the extensions found on the 80486. Later family members add additional "MMX" instructions for handling vector graphical processing, and a lot more cache memory.

Consider Motorola 68000 family processors. Again, many derivative processors now exist that can execute original 68000 integer arithmetic instructions. These derivative processors, however, are typically much faster and have far different on-chip peripherals and cache memory than the original 68000.

PA8800, Intel, are Mutually Incompatible Families

While members of the PA8800 family have basic integer instructions in common with other members of the family, most members of this family cannot directly execute even basic 8086 integer instructions. Similarly, most members of the 8086 family cannot directly execute even basic PA8800 integer instructions. These families are therefore mutually incompatible families, making it difficult to write any part of firmware that can run on both. Similarly, the Intel Itanium and PA8800 are mutually incompatible families.

Thangadurai's and Fish's Methods Require Common Familes,

Applicant's System Enables Different Families

The methods of Thangadurai and Fish are peculiarly suited for use only with processors belonging to the same family. Applicant's method has advantage in that it is suitable for use with processors that may belong to any of several mutually-incompatible families.

Thangadurai discusses a mechanism for verifying processor compatibility with firmware at system boot time. The method of Thangadurai executes on the *same* processor on which the validated firmware is to execute. As such, the processor

firmware must have at least an initial portion, or kernel, written in a way such that the initial portion is operable on all processor variations that may be used with the system. This initial portion selects is what selects processor-version-specific firmware for use during the remainder of system boot and system operation. It must therefore be written using only an instruction set subset common to at least several members, including future processors, of a processor family.

For this kernel portion of firmware to run, a *single* processor family is required by Thangadurai because only members of the same family are guaranteed to have at least some basic instructions in common.

A single processor family is clearly contemplated by Thangadurai in several other ways. For example, in col 4, line 35 to col 5 line 10, discusses speed and bug-fix differences between processor implementations as examples of systems appropriate for his system. Speed and bug-fix differences are generally fairly minor differences between members of the same family, and rarely alter basic integer-processing instructions. Further, in discussing multiprocessor systems at col 6 lines 8-20, Thangadurai states that *a single, recent, firmware version* should be compatible with "all processor models" of a system, his method seeks to find a single, recent, version of firmware that is fully compatible with the latest processor in the system..

The method of Thangadurai therefore inherently is unsuitable for validating firmware compatibility with processors of mutually incompatible processor families because these processors lack a common instruction set in which a common firmware kernel can be written. Thangadurai suitable for use on homogeneous or nearly-homogeneous (i.e. having all processors belonging to the same family and capable of running a single firmware revision) computer systems, not on a heterogeneous system.

Fish's System

Fish discusses a system having a processor (Fish 12 or 112) coupled to a memory containing multiple copies of processor-specific execution firmware (Fish 104). Fish's system has a common firmware portion (Fish 102), or kernel, that, at boot time, selects and executes an appropriate copy of the processor-specific execution firmware.

As with Thangadurai, the method of Fish executes on the *same* processor on which the validated firmware is to execute. As with Thangadurai, Fish's processor firmware kernel must be written in a way such that the kernel is operable on all types of primary processors for which processor-specific execution firmware is provided. The firmware of Fish must therefore also be written using only a microinstruction set subset common to several members of the *same* processor family.

Thangadurai Combined With Fish

When combined, Thangadurai and Fish together teach a system where an initial firmware kernel executes on a primary processor of the system, that kernel being responsible for validating and/or selecting firmware for the system. Taken together, they teach operability on a homogeneous computer system. Thangadurai and Fish taken together fail to provide any of the elements:

a) a management processor,

b) operation of a heterogeneous computer system with different compatible firmware versions enabled for different processors of the system, or

c) operability on a truly heterogeneous computer system.

Further, Thangadurai and Fish together teach away from use of a management processor, or processor-specific firmware versions, by teaching not only that a primary processor of the system should perform firmware selection and updating, but that a single recent version should suffice.

Applicants System As Claimed

In Applicant's system, determination of processor identity and firmware compatibility is performed in "a management subprocessor" (claim 1). This element appears as "a management processor" in claim 13 (former claim 8).

Further, Applicant's system, in some claims, is specifically a heterogeneous cellular system; in claim 12 the second processor receives a firmware different from that received by the first processor.

The Management (Sub)Processor in the Rejection

In rejecting claims 3 and 8, (the Examiner seems to have missed this element in Claim 1), the Examiner simply claims that "both disclose the functionality of selecting .... to do so there should have been a controller or a sub processor or a management processor or some sort of interface for making this decision."

As illustrated above, both Fish and Thangadurai both teach away from using a management (sub)processor to make this decision by teaching that this decision can and should be performed by the primary processor(s) of the system.

As illustrated above, Applicant's use of a management processor or subprocessor to make this decision provides advantage in a system by enabling function of a heterogeneous cellular system not supported by either Fish or Thangadurai.

In these circumstances, the Examiner cannot introduce the missing element of the management processors and their function into a combination of Fish and Thangadurai without finding more support in the art.

Conclusion

Applicants respectfully request that the Examiner reconsider the application as amended in light of the foregoing remarks.

It is believed that no fees are due in connection with this amendment. If any additional fee is due, please charge Deposit Account No. 08-2025.

Respectfully submitted,

By: 

Dan Cleveland, Jr., Reg. No. 36,106
LATHROP & GAGE L.C.
4845 Pearl East Circle, Suite 302
Boulder, CO 80301
Telephone: (720) 931-3012
Facsimile: (720) 931-3001